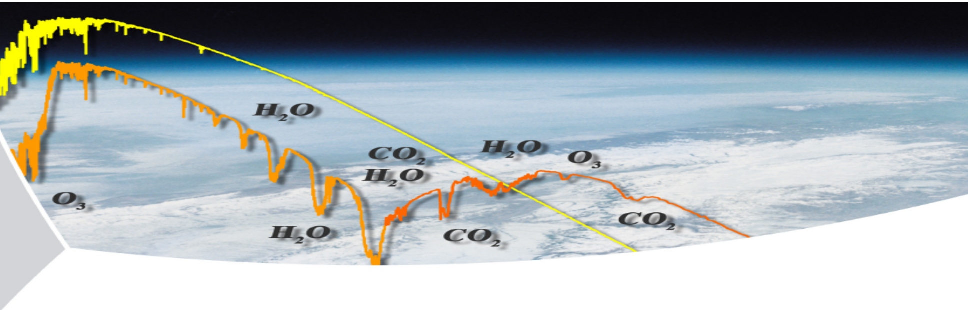


How to Tackle the Computational Challenges of Line-by-line Modelling

Franz Schreier[1]

Sebastián Gimeno García[1], Dieter Kohlert[2], . . .

[1]DLR — Remote Sensing Technology Institute
Oberpfaffenhofen, GERMANY

[2]University Regensburg, GERMANY

July 2015

1. Introduction
2. Numeric and Software
   - Voigt Function
   - Multigrid
   - Jacobians
3. Hardware
   - MultiThreading
   - Accelerators
4. Summary and Outlook

Infrared Radiative Transfer in the Atmosphere

- Schwarzschild: monochromatic intensity / radiance

$$I(\nu, s) \;=\; I(\nu, s_0)\, \mathrm{e}^{-\tau(\nu; s_0, s)} \;+\; \int_0^{\tau} \mathrm{d}\tau'\; B(\nu, T(\tau'))\; \mathrm{e}^{-\tau'}$$

- Beer: Transmission $\mathcal{T}$ and optical depth τ

$$\mathcal{T}(\nu) \;=\; \mathrm{e}^{-\tau} \;=\; \exp\left(-k(\nu)\, n\, s\right)$$

Quite simple!

... but in an inhomogeneous atmosphere with some molecules**?¿?**

$$\tau(\nu) \;=\; \int_{\text{path}} \mathrm{d}s \sum_m \sum_l S_l(T(s))\; g_{\mathrm{L}}\left(\nu; \hat{\nu}_l, \gamma_l^{\mathrm{L}}(p(s), T(s)\right) \otimes g_{\mathrm{G}}\left(\nu; \hat{\nu}_l, \gamma_l^{\mathrm{G}}(T(s)\right)\; n_m(s)$$

Lbl Challenges

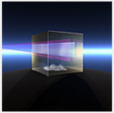
- Inhomogeneous atmosphere:
  dozens of altitude levels
- Thousand ... millions of ν grid points
- LbL–IR–RT for Remote Sensing:
  from GigaBytes to TeraBytes to ...
- Input data:
  HITRAN, GEISA, ... databases
  hundreds ... (ten)thousands of lines

  HiTemp, ExoMol, ...
  million ... billions of lines

2002-2012 MIPAS
$17 \times 72 \times 14$ spectra/day
10^9 floats/day

2006–, 2012– IASI
10^6 spectra (20GB) / day

2020 ? PREMIER
10^7 spectra/day



Lbl Challenges

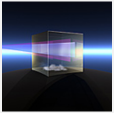
- Inhomogeneous atmosphere:
  dozens of altitude levels
- Thousand . . . millions of ν grid points
- LbL–IR–RT for Remote Sensing:
  from GigaBytes to TeraBytes to . . .
- Input data:
  HITRAN, GEISA, . . . databases
  hundreds . . . (ten)thousands of lines

  HiTemp, ExoMol, . . .
  million . . . billions of lines

2002-2012 MIPAS
$17 \times 72 \times 14$ spectra/day
10^9 floats/day

2006–, 2012– IASI
10^6 spectra (20GB) / day

2020 ? PREMIER
10^7 spectra/day



GARLIC — Generic Atmospheric Radiation Lbl Infrared Code

Line shapes:	Voigt, VanVleck$\otimes$Doppler, Lorentz
Line data:	HITRAN, HITEMP, GEISA, JPL, ...
Continua:	CKD 2.0 (H_2O, CO_2, N_2, O_2), "dry air" (Liebe)
Geometries:	Limb, uplooking, downlooking (refraction optional)
Instruments:	Spectral response: FTS, Heterodyne, Fabry–Perot, ...
	Field-of-view: Box, Gauss, Trapez, ...
Implementation:	FORTRAN 2008 with OpenMP *all data* read from external files
Inversion:	Jacobians by automatic differentation
Extensions:	(multiple) scattering infrared radiative transfer: *J. Mendrok:* SARTRE — approx. spherical geo *(GRL 2007)* *M. Vasquez:* cloudy exo-planet atmospheres *(A&A 2013a,b)*

F. Schreier et al., JQSRT, 137, 29–50, 2014

Outline

1. Introduction
2. Numeric and Software
   - Voigt Function
   - Multigrid
   - Jacobians
3. Hardware
   - MultiThreading
   - Accelerators
4. Summary and Outlook

The Voigt Function

$$K(x,y) = \frac{y}{\pi} \int_{-\infty}^{\infty} \frac{\mathrm{e}^{-t^2}}{(x-t)^2 + y^2} \, \mathrm{d}t$$

x distance to center

y ratio Lorentz/Gauss width

- Complex error function
  (Plasma dispersion function,
  Fadde(ye)va function, ...)

$$w(z) = \frac{\mathrm{i}}{\pi} \int_{-\infty}^{\infty} \frac{e^{-t^2}}{z-t} \, \mathrm{d}t$$

$z = x + \mathrm{i}y$

- Derivatives provided simultaneously:
  $w'(z) = \frac{2\mathrm{i}}{\sqrt{\pi}} - 2z\,w(z)$

The Voigt Function

$$K(x,y) = \frac{y}{\pi} \int_{-\infty}^{\infty} \frac{\mathrm{e}^{-t^2}}{(x-t)^2 + y^2} \,\mathrm{d}t$$

x distance to center

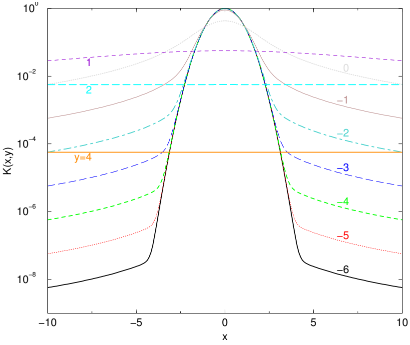
y ratio Lorentz/Gauss width

- Complex error function
  (Plasma dispersion function,
  Fadde(ye)va function, ...)

$$w(z) = \frac{\mathrm{i}}{\pi} \int_{-\infty}^{\infty} \frac{e^{-t^2}}{z-t} \,\mathrm{d}t$$

$z = x + \mathrm{i}y$

- Derivatives provided simultaneously:
  $w'(z) = \frac{2\mathrm{i}}{\sqrt{\pi}} - 2z\,w(z)$

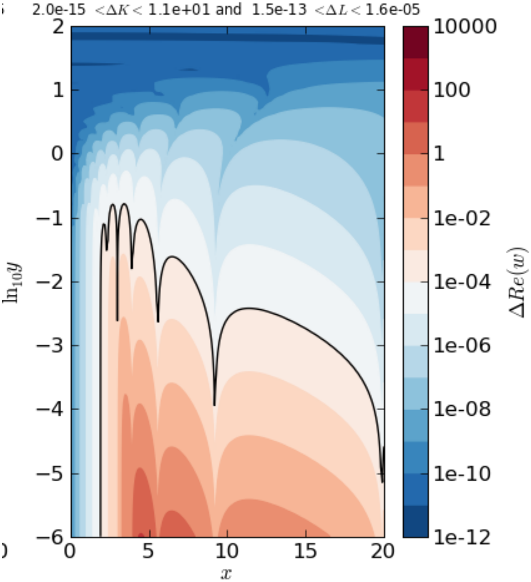


- Dozens (hundreds?) of algorithms
- Rational approximations for $w(z)$: fast (and accurate?)

Complex Error Function: Hui–Armstrong–Wray (1979)

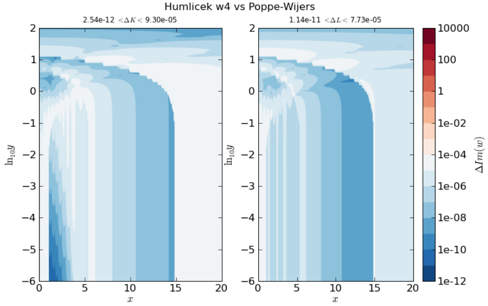
Rational approximation

$$w(z) = \frac{\sum_{m=0}^{M} a_m (\mathrm{i}z)^m}{\sum_{n=0}^{M+1} b_n (\mathrm{i}z)^n}$$



Complex Error Function: Humlicek (1982)

$$w(z) = \begin{cases} R_{6,7}(z) + \exp\left(z^2\right) & |x| + y < 5.5 \text{ and } y \geq 0.195|x| - 0.176 \\ R_{4,5}(z) & |x| + y < 5.5 \text{ otherwise} \\ R_{2,4}(z) & 5.5 \leq |x| + y < 15 \\ R_{1,2}(z) = \frac{\mathrm{i}z/\sqrt{\pi}}{z^2 - \frac{1}{2}} & |x| + y \geq 15 \end{cases}$$



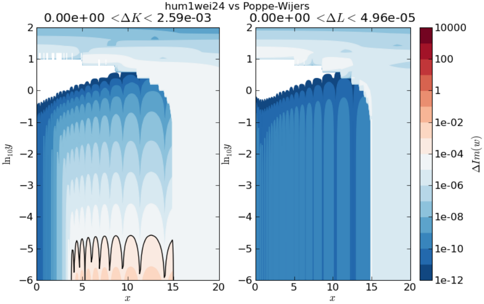
- Problem:
  efficient coding in Python
- General question:
  how many subregions?

Complex Error Function: Humlicek & Weideman

$$w(z) = \frac{\mathrm{i}}{\pi} \int_{-\infty}^{\infty} \frac{e^{-t^2}}{z-t}\,\mathrm{d}t$$ complex error function

$$= \begin{cases} \dfrac{\mathrm{i}z/\sqrt{\pi}}{z^2 - \frac{1}{2}} & \text{for } |x| + y \geq 15 \\ \dfrac{\pi^{-1/2}}{L - \mathrm{i}z} + \dfrac{2}{(L-\mathrm{i}z)^2} \sum_{n=0}^{N-1} a_{n+1} Z^n & \text{else} \end{cases}$$

$Z = (L + \mathrm{i}z)/(L - \mathrm{i}z)$
$L = 2^{-1/4} N^{1/2}$

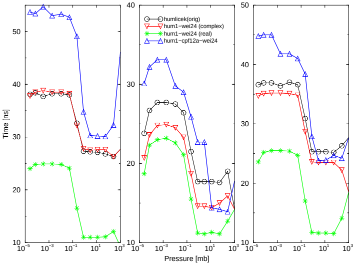


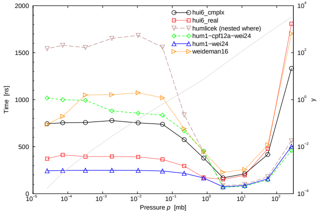
Complex Error Function: Humlicek & Weideman

$$w(z) = \frac{\mathrm{i}}{\pi} \int_{-\infty}^{\infty} \frac{e^{-t^2}}{z - t}\, \mathrm{d}t \qquad \text{complex error function}$$

$$= \begin{cases} \dfrac{\mathrm{i}z/\sqrt{\pi}}{z^2 - \frac{1}{2}} & \text{for } |x| + y \geq 15 \\ \dfrac{\pi^{-1/2}}{L - \mathrm{i}z} + \dfrac{2}{(L-\mathrm{i}z)^2} \displaystyle\sum_{n=0}^{N-1} a_{n+1} Z^n & \text{else} \end{cases}$$

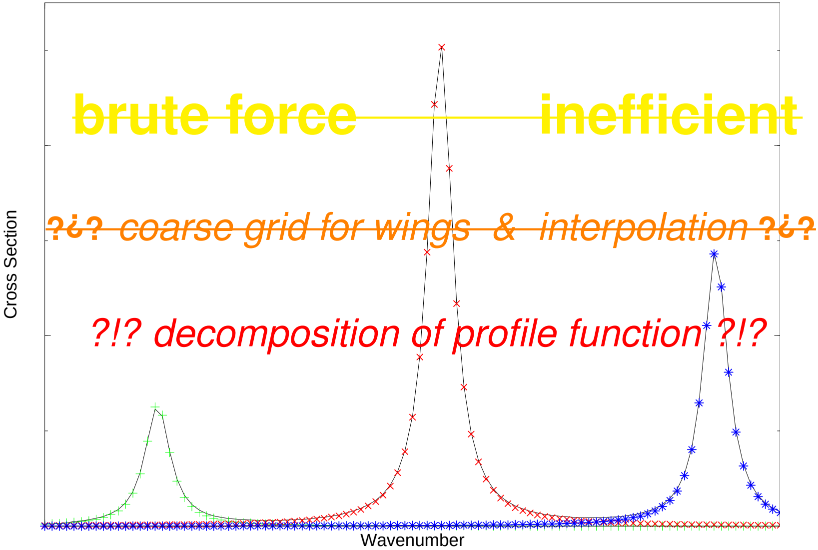
$Z = (L + \mathrm{i}z)/(L - \mathrm{i}z)$
$L = 2^{-1/4} N^{1/2}$



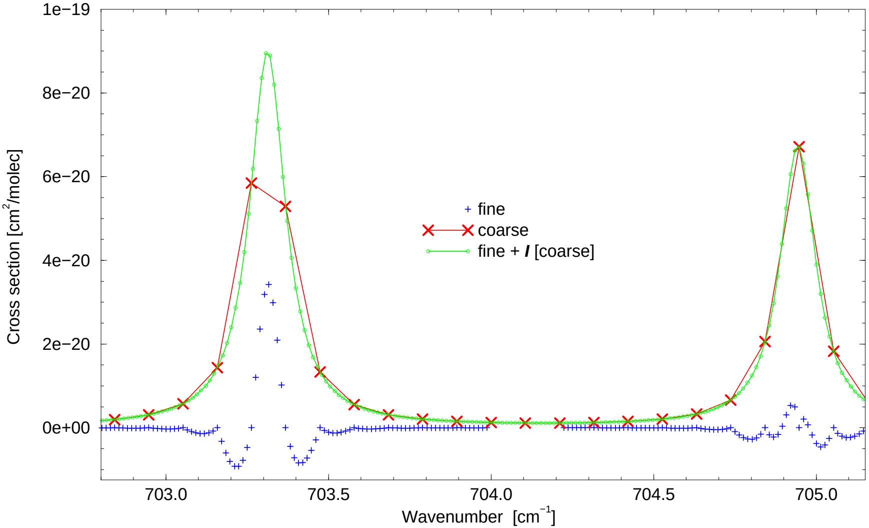


F. Schreier, JQSRT, 112, 2011

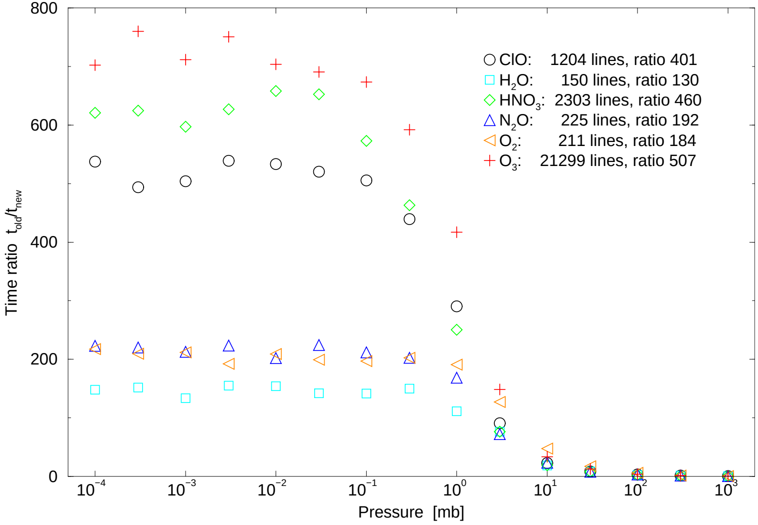
LbL Molecular Absorption Cross Sections



“Multi”-Grid Algorithm for Fast Cross Sections



"Multi"-Grid Cross Sections — MASTER $(16.5-17\,\mathrm{cm}^{-1})$



F. Schreier, CPC 2006

"Multi"-Grid: Problems and Outlook

- 2-point Lagrange interpolation:
  $+$ fast and robust
  $-$ overestimate in line wings
  $-$ underestimate in center
- 3 or 4-point Lagrange:
  $\pm$ "more" accurate?
  $-$ outliers, e.g. negative xs
  $-$ line wing cut-off difficult

New multigrid scheme under development:

- Cubic Hermite interpolation (Note: Voigt function derivatives via complex erf)
- Speed-up factor 100 . . . 500

"Multi"-Grid: Problems and Outlook

- 2-point Lagrange interpolation:
  $+$ fast and robust
  $-$ overestimate in line wings
  $-$ underestimate in center
- 3 or 4-point Lagrange:
  $\pm$ "more" accurate?
  $-$ outliers, e.g. negative xs
  $-$ line wing cut-off difficult

New multigrid scheme under development:

- Cubic Hermite interpolation (Note: Voigt function derivatives via complex erf)
- Speed-up factor 100 . . . 500

Jacobians (Derivatives, . . .)

Derivatives w.r.t. gas densities/VMR, temperature, . . . required for retrieval applications (nonlinear least squares) and sensitivity analysis

- Finite difference approximations:
  - time consuming
  - step size selection difficult
    (cancellation errors or truncation errors)
- Hand coded analytical derivatives:
  - boring and error prone
  - no “automatic update” after model upgrade
- Automatic/Algorithmic differentation
  - Kind of “precompiler”

Automatic/Algorithmic Differentation

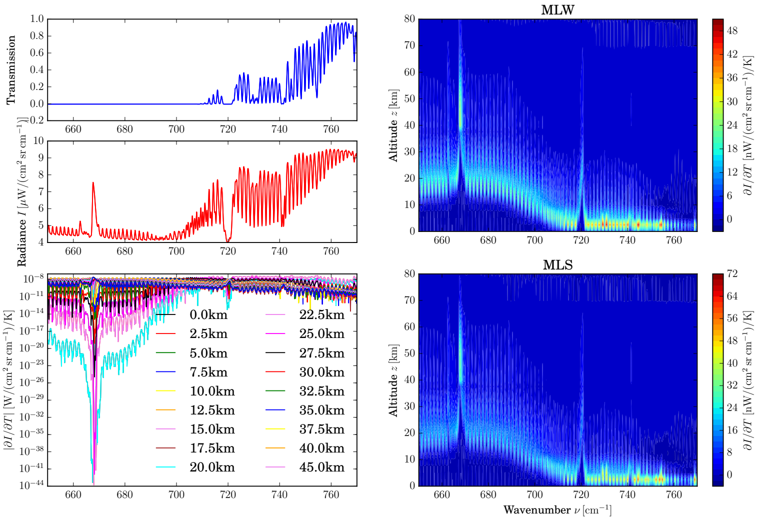
- Even large codes are essentially formulated in terms of elementary mathematical operations (sums, products, powers) and elementary functions
- Differentiation is based on simple recipes such as the chain rule (in contrast to integration)
- $\Longrightarrow$ Differentiation rules can be performed automatically by some kind of precompiler

⋆ AD generates exact derivatives

⋆ AD tools available for Fortran, C, . . .

ADIFOR (f77) or TAPENADE

Temperature Jacobians — Nadir View

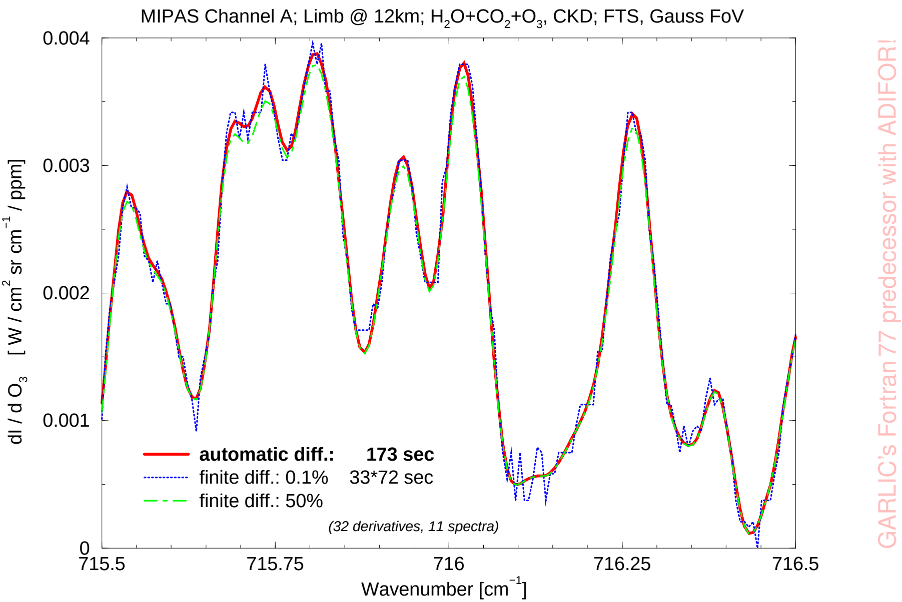


F. Schreier et al. JQSRT, 2015

$\partial I/\partial T$: 27 orders of magnitude!

$t_{\mathrm{Jac}} = 1.8 t_{\mathrm{fct}}$
(22 column Jacobian)

O_3 Derivative Spectrum @ 21km



Outline

1 Introduction

2 Numeric and Software
- Voigt Function
- Multigrid
- Jacobians

3 Hardware
- MultiThreading
- Accelerators

4 Summary and Outlook

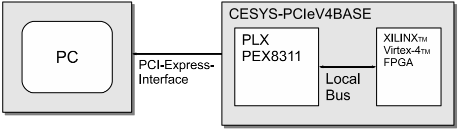
Parallelization

- Multi-core architecture of current CPUs
- Fortran 2008 with OpenMP
- in GARLIC:
  - Molecular cross sections
  - Absorption coefficients
  - Schwarzschild equation

```
DO m=1,nMolec
   readHitranGeisa
   !$ OMP PARALLEL DO
   DO j=1,nLevels
      adjustLineParameters
      DO l=1,nLines
         DO i=0,nFreqs
            xs+=strength*voigt
         END DO
      END DO
   END DO
   !$ OMP END PARALLEL DO
END DO
```

Hui-Armstrong-Wray:

$$w(z) = \frac{\sum_{m=0}^{M} a_m z^m}{\sum_{n=0}^{M+1} b_n z^n}$$



FPGA — Field Programmable Gate Arrays

D. Kohlert, U. Regensburg

XS engine:

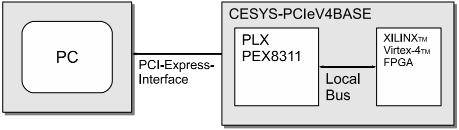
$$K(x,y) = \frac{\sum_{m=0}^{6} \rho_m(y) x^{2m}}{\sum_{n=0}^{7} \tau_n(y) x^{2n}}$$

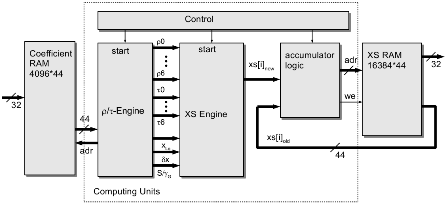
ρ and τ engines:

$$\rho_m(y) = \sum_k \phi_{m,k}(a,b)\, y^k$$

Accumulator:

$$k(\nu) += \; S \times K(x,y)$$





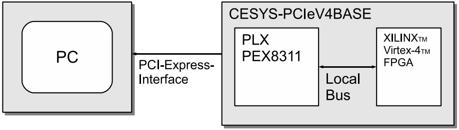
FPGA — Field Programmable Gate Arrays

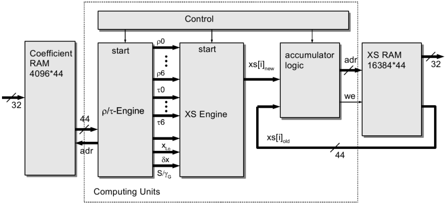
D. Kohlert, U. Regensburg

old FPGA (XILINX-Virtex):

only 48 multiplier blocks
66 MHz clock

30 ns per function value





D. Kohlert & F. Schreier, JSTARS, 2011

FPGA — Field Programmable Gate Arrays

D. Kohlert, U. Regensburg

old FPGA (XILINX-Virtex):

only 48 multiplier blocks
66 MHz clock

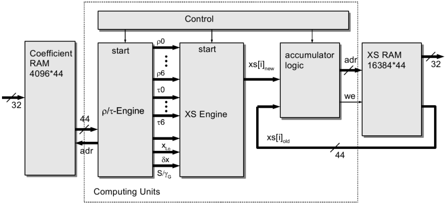
30 ns per function value

new FPGA (XILINX-Artix):

700 multipliers
250 MHz clock frequency

$<$0.5 ns

D. Kohlert & F. Schreier, JSTARS, 2011



Outline

1 Introduction

2 Numeric and Software
- Voigt Function
- Multigrid
- Jacobians

3 Hardware
- MultiThreading
- Accelerators

4 Summary and Outlook

Summary and Outlook

GARLIC

Speed-up of Lbl-RT by combination of

- Voigt function optimization
- Multigrid cross sections
- Jacobians by algorithmic differentiation
- OpenMP parallelization
- (FPGA)

Work in progress & ToDo's

- Weak line rejection
- Line wing truncation
- New multigrid scheme
- OpenMP: Convolutions
- FPGA